AD-A031 630 RAND CORP SANTA MONICA CALIF F/G 12/1

THE NUMERICAL SOLUTION OF EXPONENTIAL EQUATIONS, (U)

AUG 74 T A BROWN

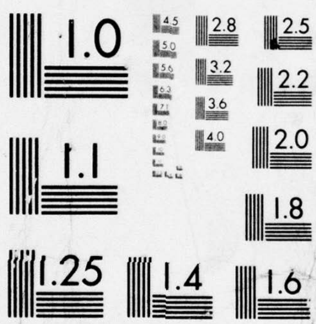
UNCLASSIFIED P-5281 NL

| OF |

AD
A031 630

END

DATE
FILMED
12-76


1.0
4.5
5.0
5.6
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

FC

2

6

THE NUMERICAL SOLUTION OF EXPONENTIAL EQUATIONS

11

August 1974

10

Thomas A. Brown

12 15p.

14 P-5281

296 600 LB

The Rand Paper Series

Papers are issued by The Rand Corporation as a service to its professional staff. Their purpose is to facilitate the exchange of ideas among those who share the author's research interests; Papers are not reports prepared in fulfillment of Rand's contracts or grants. Views expressed in a Paper are the author's own, and are not necessarily shared by Rand or its research sponsors.

The Rand Corporation
Santa Monica, California 90406

THE NUMERICAL SOLUTION OF EXPONENTIAL EQUATIONS

1. INTRODUCTION

The purpose of this note is to describe an expeditious procedure for finding the least positive root (if any) of

certain equations of the form

$$(1) \qquad \sum_{i=1}^{N} c_i \exp(-\ell_i x) = 0.$$

The quantities c_i are real constants and the quantities ℓ_i are real positive constants. This problem arises in Don Perkel's study of neuron firing subject to inhibitory and excitatory stimulation. Because the solution of equation (1) would be a small part of a monte carlo routine, it is important that the solution technique be fast and 100% reliable (that is, it should not fail catastrophically in certain circumstances, even though these circumstances might only occur once in a hundred times).

2. THE BASIC TECHNIQUE

For definiteness, let us assume that

$$0 \leq \ell_1 < \ell_2 < \ell_3 < \cdots < \ell_N.$$

Let us further assume that $c_o = 1$ (these two assumptions obviously involve no loss in generality). Now separate the terms into two categories according to whether c_i is positive or negative.

Let P be the number of positive c_i and Q the number of negative c_i; $P + Q = N$, of course, and if either P or Q were zero then there could obviously be no solution to equation (1). Let $\{(p_i, r_i): i = 1(1)P\}$ be the set of all pairs from $\{(c_i, \ell_i): i = 1(1)N\}$ such that c_i is positive, and $\{(q_i, s_i): i = 1(1)Q\}$ be the set of all such pairs which have c_i negative. Then

$$(2)\quad \sum_{i=1}^{N} c_i \exp(-\ell_i \cdot x) = \sum_{i=1}^{P} p_i \exp(-r_i \cdot x) + \sum_{i=1}^{Q} q_i \exp(-s_i \cdot x).$$

In other words, if we let

$$f(x) = \sum_{i=1}^{N} c_i \exp(-\ell_i \cdot x)$$

$$g(x) = \sum_{i=1}^{P} p_i \exp(-r_i \cdot x)$$

$$h(x) = -\sum_{i=1}^{Q} q_i \exp(-s_i \cdot x)$$

then

$$f(x) = g(x) - h(x).$$

Our problem, then, is to determine the least positive value of x such that $g(x) = h(x)$. We assume, for definiteness, that

$$0 \leq r_1 < r_2 < \cdots < r_P$$

$$0 < s_1 < s_2 < \cdots < s_Q.$$

Obviously,

$$
(3) \qquad
\begin{aligned}
&\sum_{i=1}^{Q} (-q_i) \cdot \exp(-s_1 \cdot x) \geq h(x) \\
&p_1 \exp(-r_1 \cdot x) \leq g(x).
\end{aligned}
$$

We know that $p_1 = 1$, and so if x is so large that

$$\exp((s_1 - r_1)x) > \sum_{i=1}^{Q} (-q_i)$$

we see that $h(x) < g(x)$.

<u>Theorem 1</u>: If $x > \log\left[\sum_{i=1}^{Q} (-q_i)\right] \Big/ (s_1 - r_1)$, <u>then</u> $f(x) > 0$.

Theorem 1 limits the range over which we need search for a value of x such that $g(x) = h(x)$. In order to conduct this search, we exploit the fact that both g and h are very "well-behaved" functions: they are both everywhere positive, and their n-th derivatives everywhere have the same sign as $(-1)^n$. In view of this, an easy application of the mean value theorem gives us the following:

<u>Theorem 2</u>: <u>Let</u> $G(x)$ <u>be a polynomial of degree</u> K <u>such that, for a certain real number</u> x_o,

$$
\begin{aligned}
G(x_o) &= g(x_o) \\
G'(x_o) &= g'(x_o) \\
&\vdots \\
G^k(x_o) &= g^k(x_o);
\end{aligned}
$$

<u>then if</u> $x > x_o$, $G(x) < g(x)$ <u>if</u> k <u>is odd, and</u> $G(x) > g(x)$ <u>if</u> k <u>is even. A similar result holds for</u> $h(x)$.

<u>Proof</u>: Assume k is odd. $G^{k+1}(x) = 0 < g^{k+1}(x)$. Thus $G^k(x) < g^k(x)$ for all $x > x_o$, which implies $G^{k-1}(x) < g^{k-1}(x)$ for all $x > x_o$, and so on.

Theorem 2 immediately suggests the following algorithm: if $g(0) < h(0)$, find a polynomial G of even degree such that all G's nonzero derivatives agree with those of g at 0, and find a polynomial H of odd degree such that all H's nonzero derivatives agree with those of h at 0; it is obvious from the sign of the leading coefficients that, for some $x > 0$, $H(x) = G(x)$; let x_o be the least such x, then in view of Theorem 2 we have $g(x_o) < h(x_o)$, and we repeat the process at x_o. For all $0 < x < x_o$, $g(x) < G(x) < H(x) < h(x)$, and so it is impossible for this process to "skip over" a point at which $g(x) = h(x)$. If $h(0) < g(0)$, we reverse the degrees of G and H. We terminate the process when we find an x_o such that $|g(x_o) - h(x_o)| < \epsilon$ (where ϵ is a quantity small enough that we are "willing to call it zero") or when x_o is larger than the quantity mentioned in Theorem 1 (in which case there is no positive root to (1)).

3. TIMING CONSIDERATIONS

A basic question in applying the above algorithm is deciding what degree polynomials to use in fitting g and h. There seem to be four reasonable alternatives:

(1) Use a constant to fit the lower curve and a line to fit the upper one.

(2) Use a parabola to fit the lower curve and a line to fit the upper one.

(3) Use a parabola to fit the lower curve and a cubic to fit the upper one.

(4) Use a quartic to fit the lower curve and a cubic to fit the upper one.

Use of higher order polynomials is ruled out (or at least made much more treacherous) by the impossibility of explicitly solving a polynomial of fifth degree or higher. The k-th alternative above requires the solution of a k-th degree polynomial, and it is a hard question to decide whether the smaller number of steps required by the higher order methods is counter-balanced by their increased computational complexity at each step.

As a test of the relative speed of the four methods mentioned, we considered the four functions specified in Table 1, which are each combinations of the same six exponentials.

These four functions constitute a fairly severe test for the algorithm, as can be seen from the tabulated values found in Appendix I. The speed of each method was measured in two ways: by counting the number of iterations required and by keeping track on the JOSS "timer" of the amount of time required to find the least positive root. The time-sharing feature of JOSS, and the fact that "timer" reads

Table 1

i	ℓ_i	c_i			
		I	II	III	IV
1	.1	1	1	1	1
2	.4	-28.57	-29.41	-33.33	-50
3	.5	57.14	58.82	66.67	100
4	.6	-28.57	-29.41	-33.33	-50
5	4.0	-3.00	-3.09	-3.50	-5.25
6	8.0	2.86	2.94	3.33	5
Least Positive Root:		None	4.477	3.407	.055

in hundredths of a minute, makes these times quite variable; this variableness was reduced by running the tests between 10:00 and 11:00 P.M. on a Sunday evening when there was only one other user on the machine, and by averaging five separate values for each run of methods three and four. The results are summarized in Table 2. This table makes it clear that Method 1 is terrible, Method 2 is quite good, Method 3 is best of all, and Method 4 is better than Method 2 but not quite as fast as Method 3. Therefore it appears that the optimal technique is to fit the upper curve with a cubic and the lower curve with a parabola. Experience has indicated that Fortran will be thirty to forty times faster than JOSS running at night, so it appears that Method 3 will dispose of even "hard cases" in between 100 and 200 milliseconds. An annotated JOSS program applying Method 3 is supplied in Appendix II.

Table 2

COMPARISON OF FOUR METHODS

		Function I	Function II	Function III	Function IV
Method 1	iterations	613	1135	731	100
	time	69.6 sec.	130.8 sec.	84.0 sec.	12.0 sec.
Method 2	iterations	38	27	23	5
	time	7.8 sec.	6.0 sec.	4.8 sec.	1.8 sec.
Method 3	iterations	16	12	10	2
	time	5.3 sec.	3.6 sec.	3.1 sec.	.6 sec.
Method 4	iterations	11	8	8	2
	time	5.8 sec.	3.8 sec.	4.3 sec.	1.3 sec.

APPENDIX A

Function I

x	f(x)	g(x)	h(x)
.00	.85714	61.00000	60.14286
.20	.19872	53.26204	53.06332
.40	.43413	47.86187	47.42774
.60	.61690	43.29775	42.68085
.80	.68294	39.23187	38.54893
1.00	.67741	35.56469	34.88728
1.20	.63636	32.24778	31.61142
1.40	.57976	29.24570	28.66594
1.60	.51781	26.52809	26.01028
1.80	.45565	24.06782	23.61218
2.00	.39589	21.84041	21.44453
2.20	.33989	19.82372	19.48383
2.40	.28836	17.99773	17.70936
2.60	.24161	16.34430	16.10268
2.80	.19975	14.84704	14.64729
3.00	.16272	13.49111	13.32839
3.20	.13039	12.26309	12.13270
3.40	.10255	11.15083	11.04827
3.60	.07896	10.14333	10.06437
3.80	.05932	9.23064	9.17132
4.00	.04335	8.40376	8.36041
4.20	.03074	7.65456	7.62382
4.40	.02118	6.97565	6.95446
4.60	.01438	6.36036	6.34598
4.80	.01005	5.80267	5.79262
5.00	.00789	5.29710	5.28921
5.20	.00766	4.83873	4.83107
5.40	.00909	4.42306	4.41397
5.60	.01196	4.04607	4.03411
5.80	.01605	3.70408	3.68803
6.00	.02117	3.39379	3.37262
6.20	.02712	3.11218	3.08506
6.40	.03375	2.85656	2.82281
6.60	.04091	2.62446	2.58355
6.80	.04845	2.41366	2.36521
7.00	.05627	2.22215	2.16588
7.20	.06426	2.04811	1.98385
7.40	.07232	1.88989	1.81757
7.60	.08037	1.74600	1.66563
7.80	.08834	1.61509	1.52675

Function II

x	f(x)	g(x)	h(x)
.00	.85294	62.76471	61.91176
.20	.17573	54.79974	54.62401
.40	.41864	49.24131	48.82267
.60	.60735	44.54351	43.93616
.80	.67587	40.35859	39.68272
1.00	.67072	36.58410	35.91338
1.20	.62899	33.17016	32.54117
1.40	.57124	30.08030	29.50905
1.60	.50798	27.28327	26.77529
1.80	.44448	24.75113	24.30665
2.00	.38345	22.45870	22.07525
2.20	.32629	20.38317	20.05688
2.40	.27371	18.50393	18.23023
2.60	.22604	16.80233	16.57629
2.80	.18339	15.26149	15.07809
3.00	.14572	13.86612	13.72040
3.20	.11287	12.60241	12.48955
3.40	.08464	11.45786	11.37322
3.60	.06076	10.42114	10.36038
3.80	.04095	9.48202	9.44106
4.00	.02491	8.63122	8.60631
4.20	.01232	7.86037	7.84805
4.40	.00287	7.16187	7.15900
4.60	-.00376	6.52886	6.53262
4.80	-.00786	5.95513	5.96299
5.00	-.00972	5.43506	5.44478
5.20	-.00960	4.96355	4.97316
5.40	-.00778	4.53601	4.54379
5.60	-.00449	4.14827	4.15276
5.80	.00006	3.79656	3.79650
6.00	.00565	3.47746	3.47181
6.20	.01210	3.18790	3.17580
6.40	.01924	2.92507	2.90583
6.60	.02691	2.68645	2.65954
6.80	.03498	2.46975	2.43477
7.00	.04332	2.27290	2.22958
7.20	.05183	2.09403	2.04220
7.40	.06041	1.93144	1.87103
7.60	.06898	1.78359	1.71462
7.80	.07745	1.64911	1.57165

Function III

x	f(x)	g(x)	h(x)
.00	.83333	71.00000	70.16667
.20	.06847	61.97568	61.90721
.40	.34635	55.67871	55.33236
.60	.56276	50.35708	49.79432
.80	.64291	45.61666	44.97375
1.00	.63951	41.34133	40.70183
1.20	.59460	37.47459	36.87999
1.40	.53150	33.97509	33.44360
1.60	.46209	30.80742	30.34533
1.80	.39238	27.93992	27.54754
2.00	.32541	25.34403	25.01861
2.20	.26279	22.99392	22.73113
2.40	.20532	20.86624	20.66093
2.60	.15337	18.93984	18.78646
2.80	.10707	17.19558	17.08851
3.00	.06637	15.61616	15.54979
3.20	.03110	14.18592	14.15482
3.40	.00102	12.89067	12.88965
3.60	-.02416	11.71760	11.74176
2.80	-.04477	10.65510	10.69987
4.00	-.06114	9.69267	9.75382
4.20	-.07364	8.82081	8.89445
4.40	-.08262	8.03091	8.11354
4.60	-.08843	7.31521	7.40364
4.80	-.09141	6.66665	6.75806
5.00	-.09188	6.07886	6.17075
5.20	-.09015	5.54609	5.63625
5.40	-.08652	5.06312	5.14963
5.60	-.08125	4.62521	4.70646
5.80	-.07459	4.22811	4.30270
6.00	-.06677	3.86795	3.93472
6.20	-.05802	3.54122	3.59924
6.40	-.04851	3.24477	3.29328
6.60	-.03842	2.97573	3.01415
6.80	-.02791	2.73150	2.75941
7.00	-.01711	2.50974	2.52685
7.20	-.00615	2.30833	2.31449
7.40	.00485	2.12535	2.12050
7.60	.01582	1.95905	1.94323
7.80	.02666	1.80787	1.78121

Function IV

x	f(x)	g(x)	h(x)
.00	.75000	106.00000	105.25000
.20	-.38739	92.47342	92.86082
.40	.03914	83.03768	82.99854
.60	.37326	75.06474	74.69148
.80	.50281	67.96343	67.46062
1.00	.50684	61.55958	61.05274
1.20	.44843	55.76842	55.31999
1.40	.36256	50.52796	50.16539
1.60	.26707	45.78505	45.51799
1.80	.17093	41.49224	41.32131
2.00	.07876	37.60668	37.52792
2.20	-.00707	34.08963	34.09670
2.40	-.08534	30.90605	30.99139
2.60	-.15547	28.02423	28.17970
2.80	-.21728	25.41548	25.63276
3.00	-.27085	23.05383	23.32469
3.20	-.31643	20.91580	21.23223
3.40	-.35436	18.98012	19.33448
3.60	-.38508	17.22757	17.61265
3.80	-.40908	15.64072	16.04981
4.00	-.42688	14.20385	14.63072
4.20	-.43899	12.90269	13.34168
4.40	-.44595	11.72435	12.17031
4.60	-.44829	10.65717	11.10546
4.80	-.44651	9.69058	10.13709
5.00	-.44109	8.81503	9.25612
5.20	-.43249	8.02188	8.45437
5.40	-.42115	7.30330	7.72445
5.60	-.40747	6.65222	7.05969
5.80	-.39183	6.06222	6.45405
6.00	-.37457	5.52752	5.90208
6.20	-.35600	5.04286	5.39886
6.40	-.33640	4.60351	4.93992
6.60	-.31605	4.20517	4.52122
6.80	-.29517	3.84394	4.13911
7.00	-.27396	3.51632	3.79028
7.20	-.25261	3.21912	3.47173
7.40	-.23128	2.94947	3.18074
7.60	-.21010	2.70474	2.91485
7.80	-.18921	2.48260	2.67181

APPENDIX B

```
1.1 Demand N.
1.2 Do part 8 for i=1(1)N.                      Input parameters of f(x).
1.3 Do part 2.

2.11 Do step 3.1 for i=1(1)N.
2.12 Set P=0.
2.13 Set Q=0.                                   Sort out positive and negative
2.14 Do step 3.2 for i=1(1)N.                     parts of f(x).
2.15 Type "No positive root." if Q=0.
2.16 Done if Q=0.
2.2 Set T=10*(-8)•max(i=1(1)N:d(i)).            "Zero" tolerance.
2.21 Set U=log(-sum(i=1(1)Q:q(i)))/(s(1)-r(1)).  Upper limit on solution.
2.22 Set x=0.                                   Initial value of x.
2.3 Do step 3.3 for i=1(1)P.
2.31 Do step 3.4 for i=1(1)Q.
2.32 Set D=sum(i=1(1)P:R(i))+sum(i=1(1)Q:S(i)).  Evaluate f(x).
2.33 Type x in form 1 if |D|<T.
2.34 Done if |D|<T.
2.35 Set A=-sum(i=1(1)Q:s(i)*3•S(i))/6 if D<0.
2.36 Set A=-sum(i=1(1)P:r(i)*3•R(i))/6 if D>0.
2.37 Set B=[sum(i=1(1)P:r(i)*2•R(i))+sum(i=1(1)Q:s(i)*2•S(i))]/2.
2.38 Set C=-sum(i=1(1)P:r(i)•R(i))-sum(i=1(1)Q:s(i)•S(i)).
2.39 Do part 7.                                 Go and solve cubic equation.
2.4 Set x=x+z.
2.41 Type "No positive root." if x>U.
2.42 Done if x>U.
2.43 To step 2.3.

3.1 Set d(i)=c(i)/c(1).                         Part 3 is a collection of unrelated
3.2 Do part 5-sgn(d(i)).                          steps. It is never executed as a
3.3 Set R(i)=p(i)•exp(-r(i)•x).                   part.
3.4 Set S(i)=q(i)•exp(-s(i)•x).

4.1 Set P=P+1.
4.2 Set p(P)=d(i).
4.3 Set r(P)=l(i).
                                                Sorting out positive and negative
6.1 Set Q=Q+1.                                    parts of f(x).
6.2 Set q(Q)=d(i).
6.3 Set s(Q)=l(i).

7.1 Set H=(3•A•C-B*2)/(9•A*2).
7.11 Set G=(2•B*3-9•A•B•C+27•A*2•D)/(27•A*3).
7.12 Set E=G*2+4•H*3.                           Finding the smallest positive root of
7.13 To step 7.3 if E<0.
7.14 To step 7.4 if H=0.                          AZ^3 + BZ^2 + CZ + D = 0
7.2 Set u=(-G+sqrt(E))/2.
7.21 Set u=sgn(u)•|u|*(1/3).                    by Cardano's method. Notation is
7.22 To step 7.4 if u=0.                        very close to that in Conkwright,
7.23 Set z=u-H/u-B/(3•A).                       "Introduction to the Theory of
7.24 Done.                                      Equations," pp. 68 - 77.
7.3 Set w=arg(-G,sqrt(-E)).
7.31 Set M=cos((w-6.2831853)/3) if G≥0.
```

7.32 Set M=cos(w/3) if G<0.
7.33 Set z=2•sqrt(-H)•M-B/(3•I).
7.34 Done.
7.4 Set z=sgn(-G)•|G|*(1/3)-B/(3•A).

8.1 Demand c(i). *Input of parameters of f(x).*
8.2 Demand l(i).

Form 1:
Smallest positive root: x=............